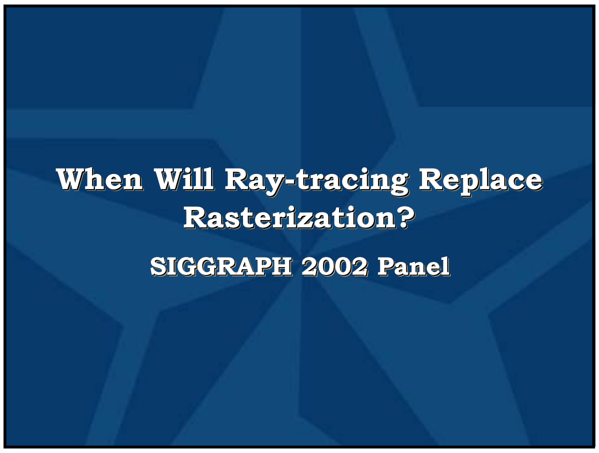
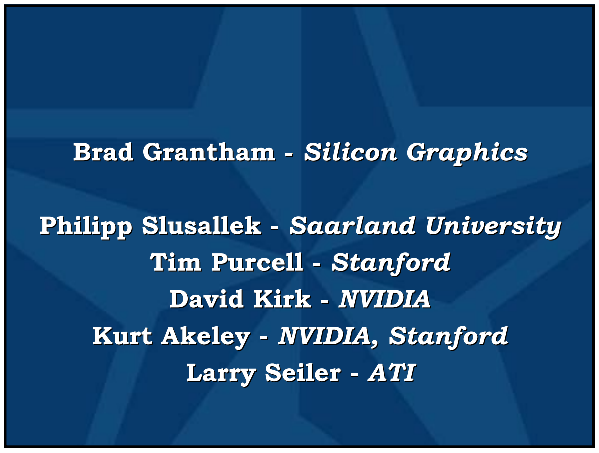


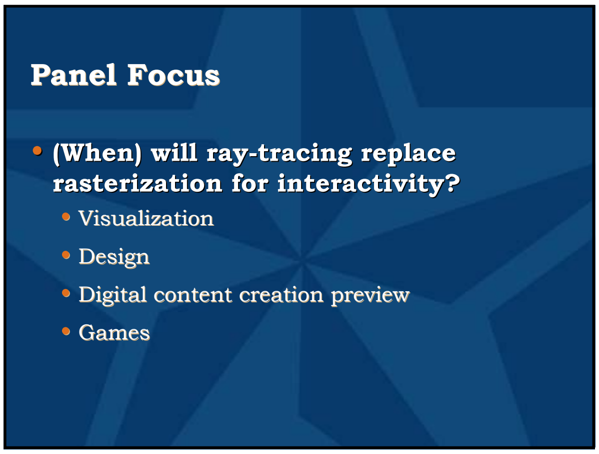
When Will Ray-tracing Replace Rasterization?

SIGGRAPH 2002 Panel



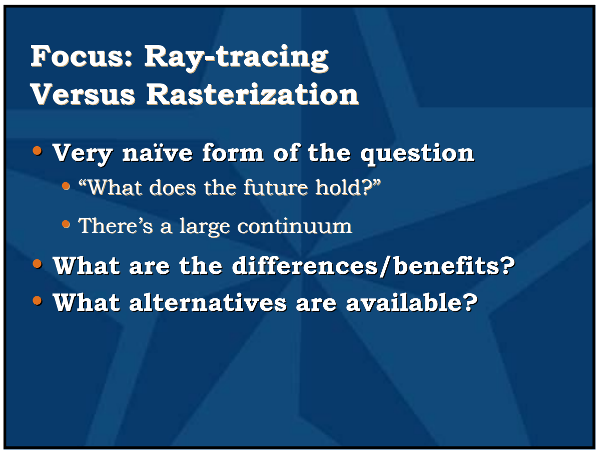
Brad Grantham - *Silicon Graphics*

Philipp Slusallek - *Saarland University*
Tim Purcell - *Stanford*
David Kirk - *NVIDIA*
Kurt Akeley - *NVIDIA, Stanford*
Larry Seiler - *ATI*



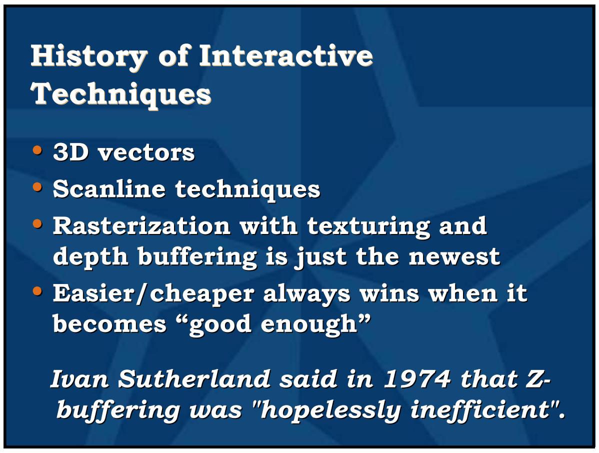
Panel Focus

- **(When) will ray-tracing replace rasterization for interactivity?**
  - Visualization
  - Design
  - Digital content creation preview
  - Games



Focus: Ray-tracing Versus Rasterization

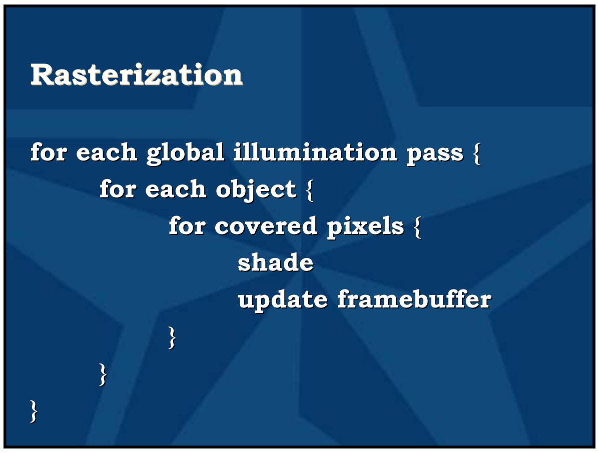
- **Very naïve form of the question**
  - "What does the future hold?"
  - There's a large continuum
- **What are the differences/benefits?**
- **What alternatives are available?**



History of Interactive Techniques

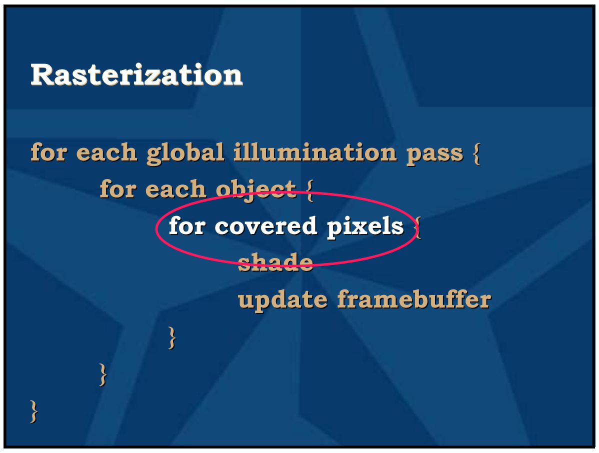
- **3D vectors**
- **Scanline techniques**
- **Rasterization with texturing and depth buffering is just the newest**
- **Easier/cheaper always wins when it becomes "good enough"**

Ivan Sutherland said in 1974 that Z-buffering was "hopelessly inefficient".



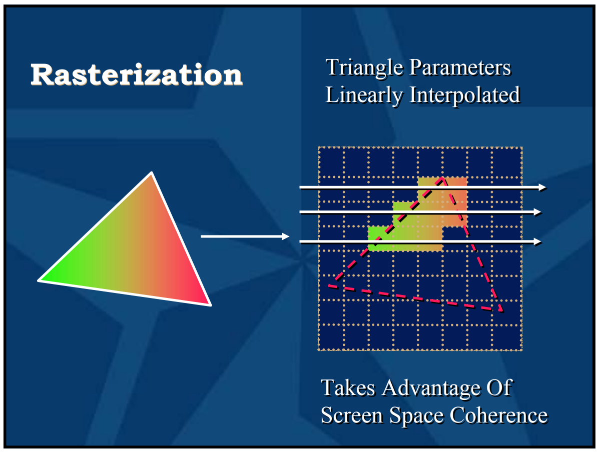
Rasterization

```
for each global illumination pass {
    for each object {
        for covered pixels {
            shade
            update framebuffer
        }
    }
}
```



Rasterization

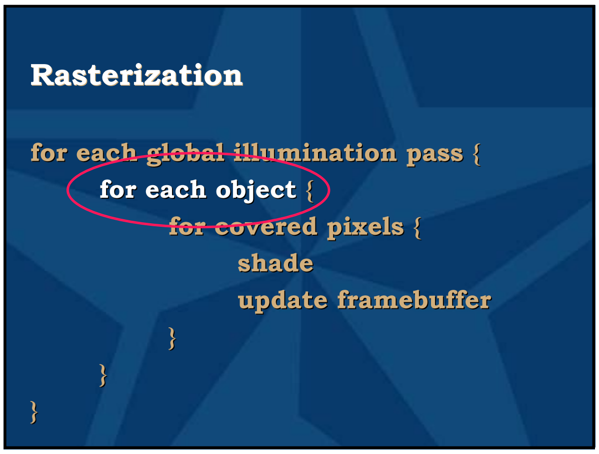
```
for each global illumination pass {
    for each object {
        for covered pixels {
            shade
            update framebuffer
        }
    }
}
```



Rasterization

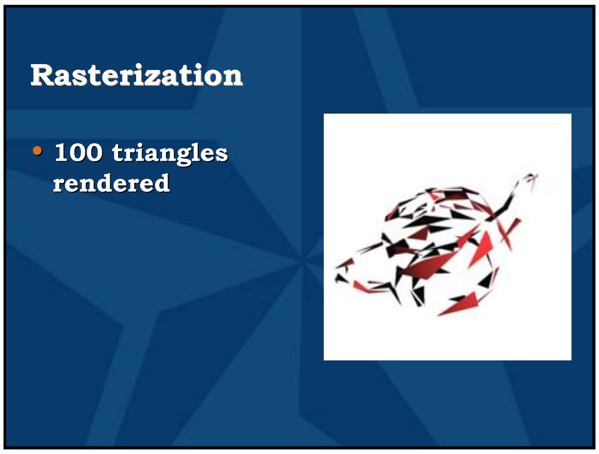
Triangle Parameters
Linearly Interpolated

Takes Advantage Of
Screen Space Coherence



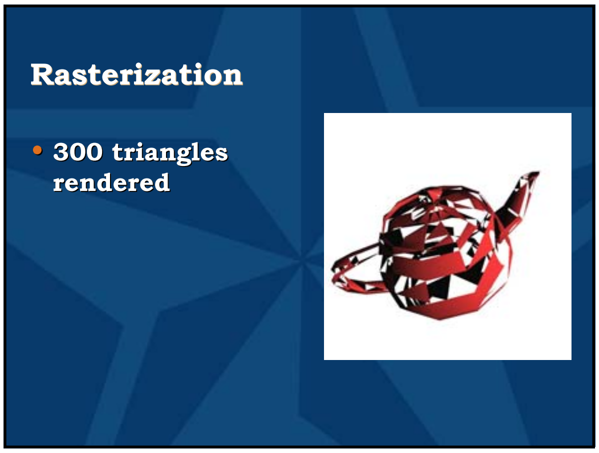
Rasterization

```
for each global illumination pass {
    for each object {
        for covered pixels {
            shade
            update framebuffer
        }
    }
}
```



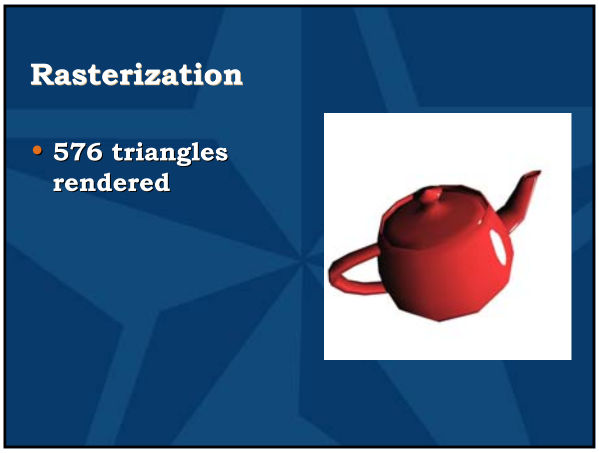
Rasterization

- 100 triangles rendered



Rasterization

- 300 triangles rendered



Rasterization

- 576 triangles rendered

Rasterization

- **Many different variations**
  - Multiple pixels at a time
  - Multiple triangles at a time
  - Could even cast rays per-pixel
- **For purpose of argument:**

  App feeds primitives one at a time

Rasterization

- **Quality can be quite high**
  - Reflections, refractions, bump-mapping
  - Various lighting models, shadows
  - Motion blur, anti-aliasing, depth-of-field
  - At what cost?

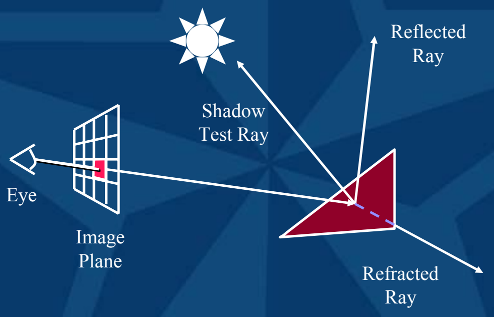
Ray-tracing

```
for all pixels {
        for each ray in path {
                find intersected object {
                        shade
                }
        }
        update framebuffer
}
```

Ray-tracing

```
for all pixels {
        for each ray in path {
                find intersected object {
                        shade
                }
        }
        update framebuffer
}
```

Ray-tracing

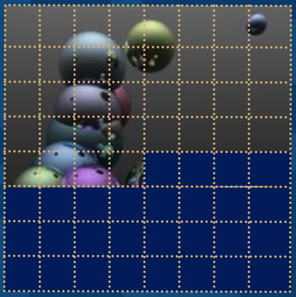


Ray-tracing

```
for all pixels {
        for each ray in path {
                find intersected object {
                        shade
                }
        }
        update framebuffer
}
```

Ray-tracing

- **Each pixel rendered fully and independently**



Ray-tracing

- **Many different variations**
  - Trace bundle of rays at a time (partially invert the loop)
  - Use Z-buffer results as first pass
- **For purpose of argument:**

  Each pixel fully evaluated

Ray-tracing

- **Complex effects easy to implement**
  - But may require significant math per intersection
- **Secondary interactions need help**
  - e.g. Caustics, color bleeding
  - Solutions exist - Photon mapping, Metropolis

Ray-tracing

- **Acceleration is necessary**
  - Hierarchy
  - Gridding, static and dynamic
  - Locality - memory coherent
  - Bundling
  - SIMD

Ray-tracing Versus Rasterization

- **Rasterization is FAST**
  - > 100 million polygons per second
  - > 1 billion pixels per second
  - Pipelined, parallelized hardware
- **Ray-tracing is SLOW (on CPU)**
  - 10s of M of raw tri intersections/sec

  Not including add texture accesses, shading…

Why Use Ray-tracing?

- **Is it simply easier to implement?**
- **Raytracing can rely on Moore's law**
  - Hardware implementation?
- **Quality of rasterization depends on cleverness**
  - Pixel shaders
  - Multipass

When Will Ray-Tracing Replace Rasterization?

SIGGRAPH 2002 Panel